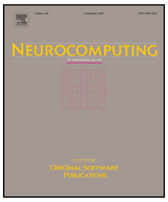
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Inferring single-cell trajectories via critical cell identification using graph centrality algorithm

Yanglan Gan [a], Jiaqi Chu [a], Guangwei Xu [a], Cairong Yan [a], Guobing Zou [b,*]

[a] *School of Computer Science and Technology, Donghua University, Shanghai, 200000, China*
[b] *School of Computer Engineering and Science, Shanghai University, Shanghai, 200000, China*

ARTICLE INFO

Communicated by G. Yu

Dataset link: GSE52529, GSE67310, http://hrpi.ddnetbio.com/, https://zenodo.org/records/1443566/files/real/gold/human-embryos_petropoulos.rds?download=1, https://www.pnas.org/highwire/filestream/29285/field_highwire_adjunct_files/1/pnas.1621412114.sd02.xlsx

Keywords:
Trajectory inference
Critical cells
Graph centrality algorithm
Greedy strategy

ABSTRACT

Trajectory inference (TI) aims to infer cell differentiation trajectories in biological processes. Numerous computational methods have been developed to infer cell lineages from single-cell gene expression data. However, cluster-based methods involve a discretization that fails to capture the continuous nature of differentiation processes, while graph-based methods directly estimate the differentiation process from gene expression profiles without detecting subpopulations, making them susceptible to noise. To address these limitations, we propose scTICG, a single-cell trajectory inference method through critical cell identification and a greedy strategy. scTICG integrates the strengths of cluster-based and graph-based methods. Initially, a cluster-based backbone structure is constructed to serve as a coarse-grained trajectory. Then, considering the dynamics of cell state transitions and the influence of certain critical cells, we identify these critical cells using the graph centrality algorithm. Subsequently, these critical cells are leveraged to refine the trajectory using a greedy strategy. We evaluate scTICG on five public datasets and compare its performance with eight state-of-the-art trajectory inference methods. The experimental results demonstrate that scTICG can infer more accurate and robust trajectories compared to competitive methods. The R code for scTICG is freely available at https://github.com/DHUDBlab/scTICG.

1. Introduction

Cells undergo state transitions during many biological processes, such as development, reprogramming, regeneration and cancer. Recent advances in single-cell sequencing technologies provide opportunities to study these cellular dynamic processes at the single-cell level [1–6]. Based on the observation that developmentally related cells tend to share similarities in gene expression, cell trajectory inference reconstructs developmental processes and orders cells along trajectories in an unbiased manner [7]. The inferred trajectories can provide researchers insights into the process of cellular decision-making and fate specification [8].

Many computational methods have been proposed for trajectory inference. On the whole, these methods can be broadly divided into two categories, including cluster-based and graph-based methods [9,10]. Cluster-based TI methods usually first cluster cells into distinct states in a low dimensional space, and subsequently connect these clusters to construct trajectory structures based on minimum spanning tree (MST) or reverse graph embedding (RGE) [11]. TSCAN [12] is a pioneering method that constructs developmental trajectories based on cell clusters. Slingshot [13] employs a simultaneous principal curve to fit the cluster-based MST and obtain a smoother trajectory, then orthogonally projects cells onto the trajectory to calculate pseudotime. TIPD [14] adopts signal entropy to quantify the heterogeneity of each cluster and identifies the cluster with the highest heterogeneity as the start cluster, simultaneously introducing Jensen–Shannon divergence (JSD) to calculate the distance between cell clusters. Monocle2 [11] initially partitions cells into clusters based on the k-means algorithm and builds a spanning tree between the centroids in the reduced space. Then, it uses RGE to iteratively adjust the spanning tree and the positions of centroids to maintain the mapping between the high-dimensional gene expression space and the reduced space. As an extension, Monocle3 [15] first utilizes UMAP to reduce the dimensionality of the data and subsequently follows the main idea of Monocle2 to infer pseudotime trajectories. Different from cluster-based methods, graph-based TI methods typically construct a cell-to-cell similarity graph and then infer trajectory topology based on the graph [10,16]. For example, Wanderlust [17] is a graph-based trajectory inference method that receives multiparameter single-cell events as input and maps them onto a one-dimensional developmental trajectory. Diffusion Pseudotime (DPT) [18] defines pseudotime as the sum of all lengths over

[*] Corresponding author.
E-mail address: gbzou@shu.edu.cn (G. Zou).

https://doi.org/10.1016/j.neucom.2025.129482
Received 18 October 2024; Received in revised form 28 December 2024; Accepted 16 January 2025
Available online 23 January 2025

diffusion-like random walks based on the transition probability matrix, and branching points are identified based on the triangle inequality principle. PAGA [19] extends DPT and can estimate pseudotime on disconnected graphs. It adopts the Louvain algorithm to partition cells, and subsequently orders cells within each connected component using a random-walk-based distance measure. BLTSA [16] utilizes tangent space to identify branch and tipping cells as well as find each cell's neighborhood local tangent coordinates and calculate the global coordinates of cells as pseudotime. Although these trajectory inference algorithms have made significant progress in inferring cellular trajectories, they still encounter some challenges. Specifically, cluster-based methods can infer more robust trajectories, but they view the process of cell development as a discrete cellular state process and ignore critical cells in the process. In contrast, graph-based methods account for the continuity of cell differentiation. However, they are susceptible to noise interference.

As cellular states undergo dynamic changes over time, certain critical cells might significantly influence the cellular process. Recently, scTite [20] combines signal entropy with a soft-clustering method to identify transition cells and construct transitional paths. MuTrans [21] utilizes multi-scale reduction to construct a dynamic manifold and identify stable and transitioning cells, as well as transition paths. DBCTI [22] connects clusters based on the transition cells. After detecting cell states using a density-based clustering method, it calculates the probability of a cell belonging to different cell states through a sampling method and identifies transition cells. Overall, these methods increasingly focus on the effect of critical cells in the cell development process, but they still exhibit some limitations. First, they usually identify transition cells in a static way, which limits the ability to capture the dynamics of cell development. Additionally, the number of transition cells usually needs to be specified, which can influence the accuracy of trajectory inference.

To address these limitations, we present scTICG, a new single-cell trajectory inference method based on critical cells and a greedy strategy. scTICG integrates the strengths of both cluster-based and graph-based methods. Initially, we construct a minimum spanning tree to connect identified cell clusters as the backbone structure of the trajectory. Recognizing that cell development is a dynamic process, we then reconstruct each cluster–cluster trajectory based on critical cells. We employ the graph centrality algorithm to dynamically identify these critical cells. According to their states during differentiation, cells are categorized as initial, terminal and intermediate state. Choosing appropriate cells as starting and end points, we reconstruct the trajectory based on the critical cells using a greedy strategy. We evaluate the performance of scTICG on five single-cell datasets. The results validate the ability of scTICG to infer various trajectory structures, including linear, single-branch, and multi-branch trajectories. Comparisons with existing methods show that scTICG outperforms state-of-the-art methods with different assessment criteria.

2. Material and methods

2.1. The overview of scTICG

To infer cell trajectories from single-cell data, we propose scTICG, a novel single-cell trajectory inference method based on critical cells and a greedy strategy. scTICG integrates the strengths of cluster-based and graph-based methods. As illustrated in Fig. 1, the trajectory inference process of scTICG consists of three main steps. First, we construct a minimum spanning tree in the embedding space to connect all identified cell clusters, forming the initial coarse-grained trajectory. Secondly, considering the dynamics of the cell state transition, we identify critical cells and refine each cluster–cluster trajectory. This step involves two sub-steps. In the first sub-step, we employ the graph centrality algorithm to identify critical cells. We model a complete graph of cells from the two directly connected clusters and prune noise edges based on the cosine similarity between cell connections and the cluster–cluster trajectory. We then combine the refined graph with graph centrality algorithms to identify critical cells: initial cells with low in-degree centrality, terminal cells with low out-degree centrality, and intermediate cells with high closeness centrality. The candidate cells for trajectory reconstruction consist of critical cells and cluster centroids. In the second sub-step, choosing appropriate cells as starting and end points, we reconstruct the trajectory using a greedy strategy. In the third step, we employ the principal curve algorithm to smooth the reconstructed trajectory and project cells onto the trajectory to calculate pseudotime.

Step 1. Constructing the initial cluster-based trajectory

Given the preprocessed single-cell gene expression data D_{N*M}, where N and M respectively correspond to the numbers of cells and genes, we first employ dimensionality reduction techniques to project the high-dimensional data into a d-dimensional space D_{N*d}. This step reduces the computational cost of subsequent analyses and mitigates the impact of noise on the accuracy of inferred trajectories. To select the most appropriate dimensionality reduction method, we compare four different dimensionality reduction methods, including PCA [23], t-SNE [24,25], UMAP [26] and PAHTE [27]. We respectively apply these four dimensionality reduction methods to four real scRNA-seq datasets (HSMM, Fibroblast, HE and iPSC), and compare the clustering results. The comparative analysis indicates that t-SNE is more suitable for small datasets, while UMAP is more effective for large datasets (Supplementary Table 1). Consequently, we choose t-SNE as the dimensionality reduction method for datasets with $N < 2000$, otherwise, we use UMAP.

After dimensionality reduction, we cluster cells into k subpopulations, where k can be specified by the user or automatically determined by the silhouette coefficient [12,28]. To choose the proper clustering method, we conduct a comparative analysis involving hierarchical clustering, two classical methods (k-means and Gaussian Mixture Models (GMM)), and a recently developed algorithm, SLNMF [29], across four real scRNA-seq datasets. The results indicate that hierarchical clustering surpasses the other three methods in terms of accuracy (Supplementary Table 2). Furthermore, scTICG exhibits robust performance across different numbers of clusters (Supplementary Figure 2). Therefore, we utilize hierarchical clustering in the subsequent analysis. We calculate the distance between cell clusters, and subsequently construct an MST to connect all these cell clusters. Accordingly, we obtain cluster-level trajectory. For a particular trajectory segment, it is a set of ordered clusters.

Step 2. Reconstructing each cluster–cluster trajectory

Cluster-based trajectory analysis regards cellular processes as discrete states. However, driven by various molecular and environmental factors, cellular states usually undergo dynamic changes over time which are significantly influenced by some critical cells. Specifically, the centroid of the initial cluster often does not represent the true starting point of cell development, nor does the centroid of the final cluster represent the endpoint. Additionally, certain intermediate state cells serve as connectors, facilitating transitions between cell states. Therefore, we classify critical cells into three categories, including initial, terminal and intermediate cells. Intermediate cells connect initial and terminal cells along the cell state transition trajectory. We refine the trajectory based on these critical cells. Considering the complexity of trajectory reconstruction, we separately reconstruct each cluster–cluster trajectory. First, we identify critical cells using the graph centrality algorithm. These critical cells and cluster centroids are served as the candidate cells for trajectory reconstruction. Then, choosing appropriate candidate cells as starting and ending points, we reconstruct the trajectory using a greedy strategy.

Step 2.1. Identifying critical cells

Let C_i and C_j represent two directly connected clusters along a trajectory segment. We initially construct a complete directed cell-to-cell graph $G(V, E)$ for all cell nodes V in cluster C_i and C_j, where E represents the edge set and the edge weights are calculated by the

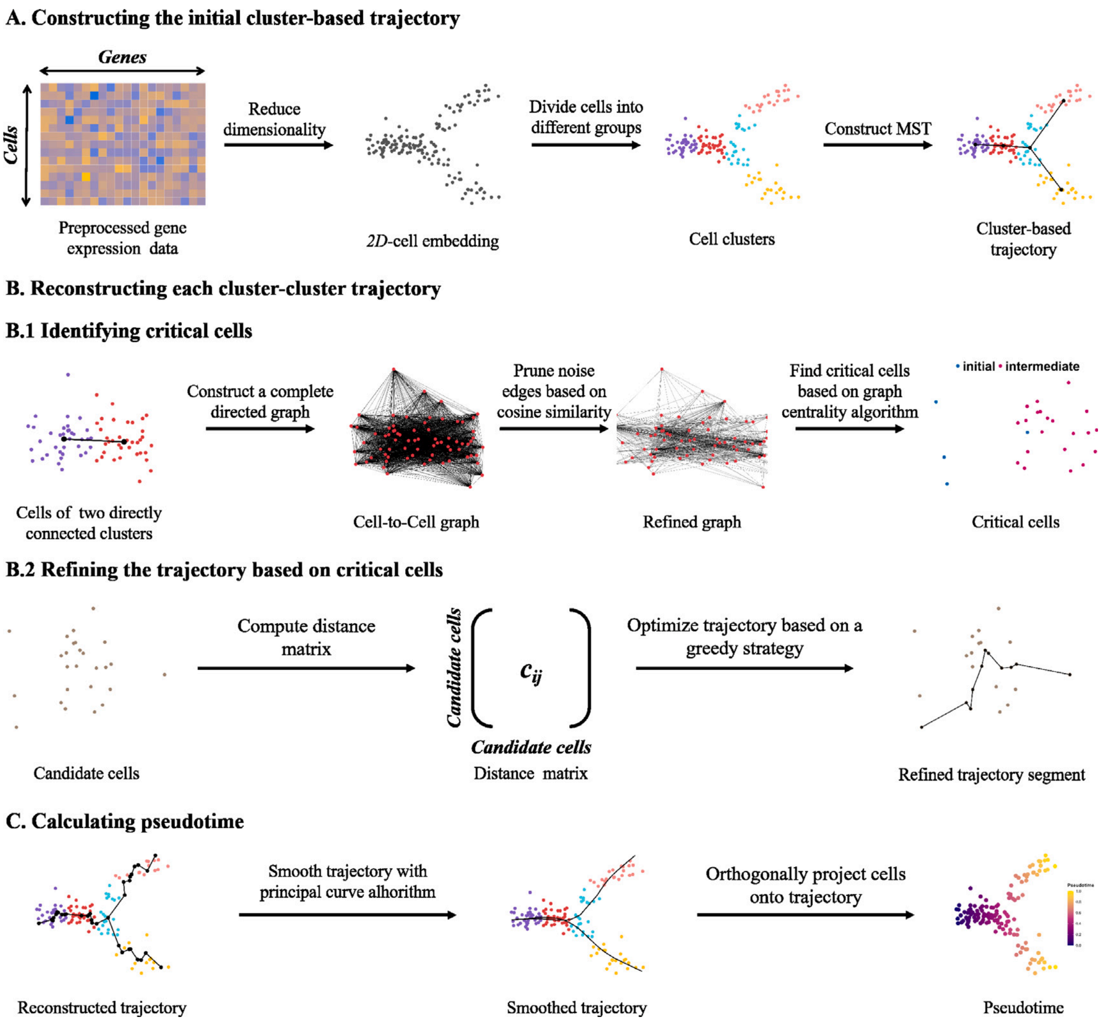

A. Constructing the initial cluster-based trajectory
Genes
Cells
Reduce dimensionality
Divide cells into different groups
Construct MST
Preprocessed gene expression data
2D-cell embedding
Cell clusters
Cluster-based trajectory
B. Reconstructing each cluster-cluster trajectory
B.1 Identifying critical cells
Construct a complete directed graph
Prune noise edges based on cosine similarity
Find critical cells based on graph centrality algorithm
• initial • intermediate
Cells of two directly connected clusters
Cell-to-Cell graph
Refined graph
Critical cells
B.2 Refining the trajectory based on critical cells
Compute distance matrix
Candidate cells
c_ij
Optimize trajectory based on a greedy strategy
Candidate cells
Candidate cells
Distance matrix
Refined trajectory segment
C. Calculating pseudotime
Smooth trajectory with principal curve alhorithm
Orthogonally project cells onto trajectory
Reconstructed trajectory
Smoothed trajectory
Pseudotime
Pseudotime

Fig. 1. The framework of scTICG. (A) Given the processed gene expression data, map the data into 2-dimensional space, use hierarchical clustering to divide cells into different states, and then construct an MST to connect all clusters. (B) Reconstruct each cluster–cluster trajectory. (B.1) Construct a complete graph for cells from the two directly connected clusters, prune noise edges based on the cosine similarity between cell connections and the cluster–cluster trajectory, and employ the graph centrality algorithm to identify critical cells (initial, terminal, and intermediate cells). (B.2) Construct a distance matrix among the candidate cells (critical cells and cluster centroids) for trajectory reconstruction, then refine the cluster–cluster trajectory based on a greedy strategy. (C) Utilize the principal curve algorithm to smooth the refined trajectory, and then project cells orthogonally onto the smoothed trajectory to calculate pseudotime.

Euclidean distance. Then, we utilize the cell-to-cell graph to identify critical cells. Considering noise edges in the graph complicates the identification of critical cells and increase computational complexity, we prune these edges based on the cosine similarity between cell connections and the cluster-level trajectory, which is calculated as:

$$\cos\theta = \frac{(c_i - c_j)\cdot(u - v)}{\|c_i - c_j\| \cdot \|u - v\|} \tag{1}$$

where c_i and c_j respectively denote the cluster centroid of cluster C_i and C_j, u and v are cell vectors of two connected cells. If $\cos\theta < 0$, indicating that the connection from the cell u to v contradicts the overall direction, we remove it from G.

In graph theory, degree centrality measures the total number of direct connections a node has with other nodes. According to the previous study [30], starting nodes typically have lower in-degree and higher out-degree, whereas terminal nodes exhibit the opposite pattern. Consequently, we utilize in-degree and out-degree metrics to identify starting and terminal cells. Previous research [31] suggests that closeness centrality and betweenness centrality are effective for identifying critical intermediate cells. Based on our comparative analysis of closeness centrality, betweenness centrality and a combination of the two (Supplementary Table 3), closeness centrality is more effective in identifying critical intermediate cells. Therefore, we utilize closeness centrality in the subsequent analyses. Based on the pruned graph $G'(V, E')$, we adopt the graph centrality algorithm [31] to calculate the in-degree, out-degree and closeness centrality of each cell. For a node v, the in-degree, out-degree and closeness centrality are calculated as:

$$\mathrm{InDegree}(v) = \frac{\sum_{u\in V \wedge u\neq v} In_{u,v}}{N-1} \tag{2}$$

$$\mathrm{OutDegree}(v) = \frac{\sum_{u\in V \wedge u\neq v} Out_{v,u}}{N-1} \tag{3}$$

$$\mathrm{CCentrality}(v) = \frac{N-1}{\sum_{u\in V \wedge u\neq v} d(u,v)} \tag{4}$$

where N is the total number of nodes in the pruned graph G'. $In_{u,v}$ denotes a directed edge from node u to node v, $Out_{v,u}$ is just the opposite. $d(u,v)$ represents the geodesic distance between nodes u and v.

Combining the three types of centrality values, we can identify critical cells $\mathrm{C}_{\mathrm{critical}}$, including initial, terminal and intermediate cells. Specifically, cells in the initial cell cluster with low in-degree centrality are identified as initial cells, while cells in the terminal cell cluster with low out-degree centrality are identified as terminal cells [30]. Closeness centrality evaluates the significance of nodes from a global graph structure perspective [32,33]. A node v with a lower average geodesic distance to all other nodes demonstrates higher closeness centrality, indicative of a robust capability to disseminate information across the graph. To automatically identify intermediate cells, we focus on cells that exhibit higher closeness centrality. Utilizing the GMM in the mclust package [34], cells are automatically categorized into L levels, denoted as $\{level_1, level_2, \dots, level_L\}$. The efficiency of information transmission increases from $level_1$ to $level_L$. Consequently, cells at $level_L$ are recognized as intermediate cells. For all cells in V, initial, terminal and intermediate cells are identified as below:

$$\mathrm{C}_{\mathrm{initial}} = \{v \in V : v \in \mathrm{Cluster}^{\mathrm{initial}} \wedge \mathrm{InDegree}(v) < \delta\} \tag{5}$$

$$\mathrm{C}_{\mathrm{terminal}} = \{v \in V : v \in \mathrm{Cluster}^{\mathrm{terminal}} \wedge \mathrm{OutDegree}(v) < \delta\} \tag{6}$$

$$\mathrm{C}_{\mathrm{intermediate}} = \{v \in V : v \in level_L\} \tag{7}$$

where $\mathrm{Cluster}^{\mathrm{initial}}$ denotes the initial cluster, $\mathrm{Cluster}^{\mathrm{terminal}}$ represents the terminal cluster, and δ is a threshold. Here, according to the performance analysis of different thresholds across four scRNA-seq datasets (Supplementary Figure 3), we set the default value of δ to 0.05. As the computational complexity increases with the number of cells increases, for larger clusters, we randomly select M (defaulted value: 200) cells from the cluster to find critical cells rather than analyzing all cells in the cluster.

Step 2.2. Refining the trajectory based on critical cells

Based on these identified critical cells $\mathrm{C}_{\mathrm{critical}}$, we can refine the trajectory from cluster C_i to C_j. Each trajectory reconstruction is initialized at a starting node and terminated at any ending node. Here, the starting and ending nodes of trajectory reconstruction are defined as:

$$\mathrm{Node}_{\mathrm{start}} = \begin{cases} \mathrm{C}_{\mathrm{initial}} & \text{if } \mathrm{C}_{\mathrm{initial}} \neq \emptyset \\ center_i & \text{otherwise} \end{cases} \tag{8}$$

$$\mathrm{Node}_{\mathrm{end}} = \begin{cases} \mathrm{C}_{\mathrm{terminal}} & \text{if } \mathrm{C}_{\mathrm{terminal}} \neq \emptyset \\ center_j & \text{otherwise} \end{cases} \tag{9}$$

where $center_i$ and $center_j$ respectively denote the centroid of cluster C_i and C_j.

Thus, we obtain the candidate set for trajectory reconstruction, $\mathrm{Node}_{\mathrm{start}} \bigcup \mathrm{C}_{\mathrm{intermediate}} \bigcup \mathrm{Node}_{\mathrm{end}}$. Taking each node in $\mathrm{Node}_{\mathrm{start}}$ as the starting point, we iteratively reconstruct the trajectory based on a greedy strategy. We construct a distance matrix D among these candidate nodes and prune noise edges to get D'. The node $node_{kt}$ at step k of the tth trajectory reconstruction is selected from candidate nodes to the reconstructed path sequence p_t if $node_{kt}$ is closest to $node_{(k-1)t}$ and $node_{kt} \notin p_t$ according to D'. In the tth trajectory reconstruction, after k iterations, we get a sequence of nodes on the reconstructed path, representing as $p_t = \{node_{1t}, \dots, node_{kt}\}$. Eventually, we obtain multiple reconstructed paths and then choose the path that covers most cells as the final reconstructed path from the pool of candidate reconstructed paths.

Step 3. Calculating pseudotime

Based on the reconstructed cluster–cluster connections, we get the refined trajectory. To obtain the final smoothed development trajectory, we utilize the principal curve algorithm [35] to fit it. Subsequently, we orthogonally project cells onto the smoothed trajectory, calculating the distance from the projection point of each cell to the starting cell along the trajectory. These distances represent the pseudotime values of the cells, denoted as $\pi = \{\pi_i\}_{i=1}^{N}$. We then rescale the pseudotime values using a max–min scale to ensure they fall within the range of 0 and 1. The calculation is formulated as:

$$\pi_i' = \frac{\pi_i - \min\pi}{\max\pi - \min\pi} \tag{10}$$

2.2. Datasets and data preprocessing

We evaluate the performance of scTICG on five widely-used datasets, including four real single-cell RNA sequencing (scRNA-seq) datasets (including HSMM, Fibroblast, HE and iPSC) and one synthetic dataset (Hrpi). The HSMM dataset contains 372 cells from differentiating human skeletal muscle myoblasts and is collected at four different time points (0, 24, 48 and 72 h) [36]. The Fibroblast dataset encompasses 355 cells sampled at five distinct time points (0, 2, 5, 20 and 22 days) during the differentiation process from mouse embryonic fibroblasts into induced neuronal cells [37]. The HE dataset consists of 1289 human embryo samples collected at five distinct time points (3, 4, 5, 6 and 7 days) [7]. The iPSC dataset is induced pluripotent stem cell [21], containing 1896 cells and 96 genes after removing heterogeneous cells for quality control and collected at 0, 1, 1.5, 2, 2.5, 3, 4 and 5 days. The Hrpi dataset integrates scRNA-seq and snRNA-seq data, containing 43791 cells and 11549 genes [38].

The inherent sparsity and noise in single-cell data necessitate rigorous preprocessing. Here, Our data preprocessing encompasses gene filtering, data normalization, feature selection and data imputation. Initially, we eliminate genes not expressed in more than 10 cells to reduce noise. Then, we apply log2 normalization to the remaining genes to mitigate technical and biological variations across single-cell samples. Subsequently, we employ feature selection techniques to reduce data dimensionality. Specifically, we select the top v (ranging from 10% to 20%) of genes exhibiting the highest variance across all cells as representative genes. To determine the optimal gene filtering ratio v, we employ clustering methods to the three filtered datasets, and assess the clustering performance. The value yielding the highest average ARI is selected for further analysis. The iPSC dataset, already preprocessed, does not require further gene filtering Finally, we utilize imputation algorithms to alleviate the impact of dropout events on subsequent analysis. On these real scRNA-seq datasets, we compare four popular imputation methods, including MAGIC [39], DrImpute [40], ALRA [41] and scISR [42]. As MAGIC consistently outperforms other methods in data imputation, it is chosen for subsequent analysis. For the Hrpi dataset, we utilize Seurat from the original literature to ensure consistency with previous analyses [38].

2.3. Evaluation metrics

We employ four methods to comprehensively evaluate the performance of trajectory inference methods. First, the pseudo-temporal ordering score (POS) [12] and Spearman's rank correlation coefficient (SRCC) are adopted to quantitatively assess the accuracy and consistency between inferred trajectories and true development trajectories. Secondly, we assess the robustness by calculating SRCC under different perturbations, including different cell sampling rates and different gene filtering ratios. Thirdly, we compare the running time under various data sizes to evaluate the efficiency of different TI methods. Finally, we evaluate and compare the accuracy of different methods in identifying biological marker genes involved in the biological process.

It is assumed that external information not used in pseudotime inference is available to evaluate the pairwise order of cells. Let π denotes an inferred path of N cells, the POS is calculated as the sum of $g(\pi, i, j)$ over all pairs of cells:

$$\mathrm{POS} = \sum_{i=1}^{N_\pi - 1} \sum_{j:j>i} g(\pi, i, j) \tag{11}$$

Here, $g(\pi, i, j)$ is a score that quantifies how well the order of the ith and jth cells in the ordered path matches their reference order. Considering the ith cell is precedes the jth cells in the ordered path π, $g(\pi, i, j)$ is calculated as:

$$g(\pi, i, j) = \begin{cases} 0 & \text{if } T(i) = T(j) \\ (T(j) - T(i))/D_\pi & \text{otherwise} \end{cases} \tag{12}$$

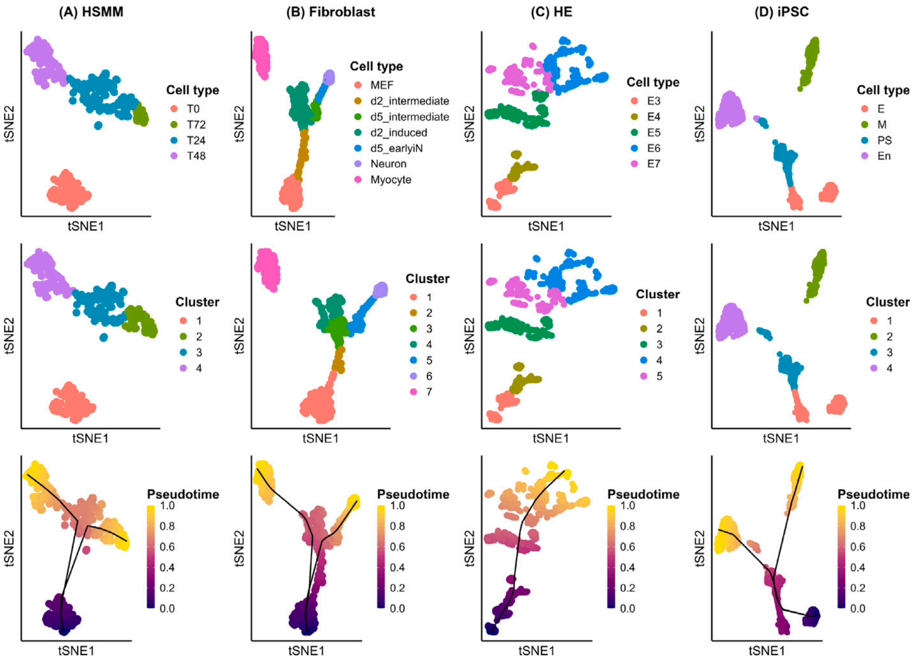


Fig. 2. The identified clusters and inferred trajectories by scTICG on fours real scRNA-seq datasets. (A) HSMM, (B) Fibroblast, (C) HE and (D) iPSC. (First row: the cell types from the original study; second row: the cell clusters identified by scTICG; third row: the pseudotime inferred by scTICG).

where $T(i)$ and $T(j)$ respectively indicate the collection time points of the ith and jth cells. D_π serves as a normalization parameter to ensure that the POS remains within the range of −1 to 1.

The SRCC is calculated as:

$$\mathrm{SRCC} = \frac{\sum_i (True_i - \overline{True})(Infer_i - \overline{Infer})}{\sqrt{\sum_i (True_i - \overline{True})^2}\sqrt{\sum_i (Infer_i - \overline{Infer})^2}} \tag{13}$$

where $True$ and $Infer$ represent the rank of the real cell ordering and the rank of the inferred pseudotime ordering, respectively. A higher SRCC value indicates that the inferred pseudotime ordering is more consistent with the true order.

3. Results

To evaluate the performance of scTICG for inferring cell trajectories, we apply it to five single-cell datasets (HSMM, Fibroblast, HE, iPSC and Hrpi), which contain pseudo-time information derived from prior studies. Additionally, we conduct a comparative analysis of scTICG against eight state-of-the-art trajectory inference methods:

- **TSCAN** [12] constructs MST based on cluster centroid and maps cells to the MST to calculate pseudotime.
- **Slingshot** [13] constructs a cluster-based MST in the embedding space, fits the MST using simultaneous principal curves, and calculates pseudo-time by orthogonally projecting cells onto the trajectory.
- **DensityPath** [43] calculates the density of cells and selects high-density clusters as representative cell states (RCSs). Then, an MST is constructed between RCSs to represent the state transition path on a density landscape.
- **TIPD** [14] introduces JSD to measure the distances between different clusters and construct MST and then uses the simultaneous principal curve to calculate pseudotime.
- **Monocle3** [15] utilizes UMAP to reduce data dimensionality and infers the pseudotime trajectories with RGE in the embedding space.
- **scTite** [20] identifies transition cells with higher transition entropy, and then constructs transitional paths to optimize the cluster-based MST.
- **PseudoGA** [44] introduces a genetic algorithm to order cells with the assumption that gene expressions vary according to a smooth curve along the trajectory.
- **BLTSA** [16] identifies tip and branching cells, clusters cells to obtain the local linear coordinates, and then assigns cells to a global trajectory by minimizing differences between local and global coordinates.

3.1. Trajectory inference on five single-cell datasets

We first assess the performance of scTICG in inferring cell trajectories on the four real scRNA-seq datasets, and detailed experimental results are shown in Fig. 2. According to previous studies [11,12,20], the HSMM dataset comprises two branches, including the primary path and a secondary path influenced by contaminating interstitial mesenchymal cells. scTICG identifies four distinct cell states and successfully infers two trajectories that closely align with the reference trajectory. For the Fibroblast dataset, scTICG infers two bifurcated trajectories, consistent with the previous study [37]. The HE dataset contains one linear trajectory from embryonic day 3 to embryonic day 7 [7]. scTICG divides cells into five clusters and successfully infers one linear trajectory. According to the previous study [21], the iPSC dataset features two distinct developmental trajectories, with branching occurring as cells transition from a primitive streak state to either a mesodermal state or an endodermal state. scTICG accurately identifies four clusters and infers the branching trajectory consistent with prior research.

Subsequently, we compare the trajectory accuracy inferred by scTICG against other competing TI methods, adopting POS and SRCC as the evaluation metrics. Each method is executed 10 times, and then the average score is calculated. To quantitatively compare our method with other competing methods, we calculate the 95% confidence intervals (CI) for both POS and SRCC metrics for each method. Additionally, we perform the Wilcoxon rank-sum test to evaluate the statistical significance of differences in POS and SRCC values between scTICG and the competing methods, with the alternative hypothesis that the POS and SRCC of scTICG are higher. A p-value less than 0.05 is considered significant, indicating that scTICG outperforms the competing methods.

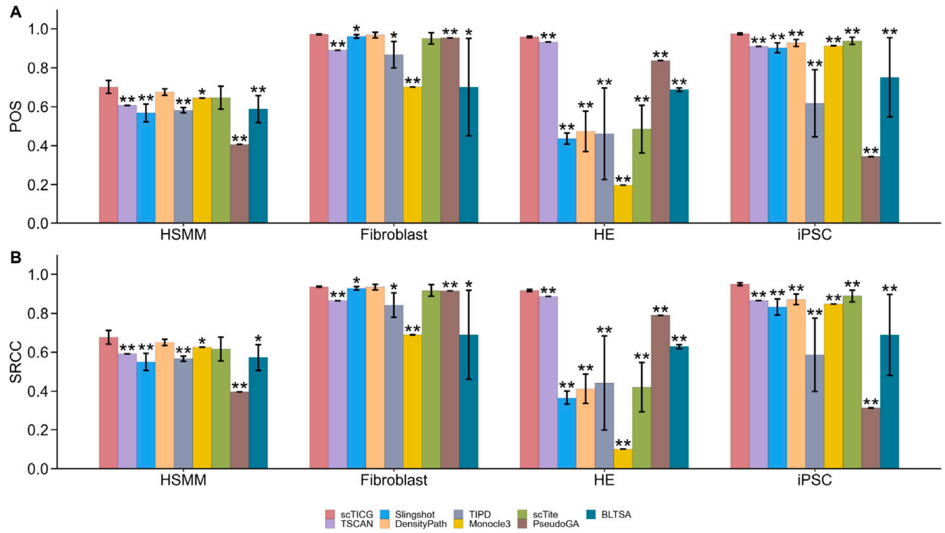


Fig. 3. The performance comparison of scTICG with eight state-of-the-art trajectory inference methods. The accuracy is measured by POS and SRCC. Error bars represent the 95% confidence intervals around the means. Statistical significance is indicated by *P< 0.05 and **P< 0.01, based on the Wilcoxon rank-sum test comparing the POS and SRCC of scTICG with each of the other methods.

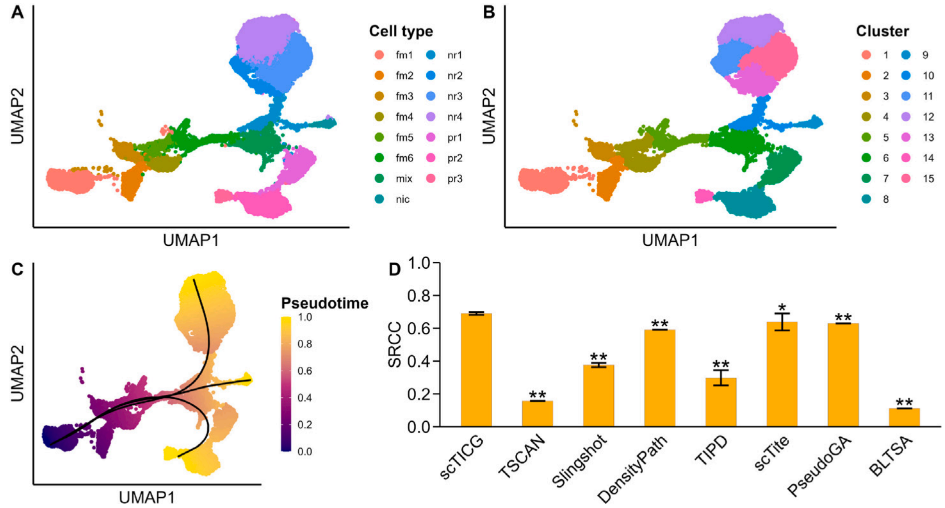


Fig. 4. Trajectory inference on the Hrpi dataset. (A) The cell types from the original study. (B) the cell clusters identified by scTICG. (C) The pseudotime trajectory inferred by scTICG. (D) Accuracy comparison between scTICG and other TI methods using SRCC. Error bars represent the 95% confidence intervals around the means. Statistical significance is indicated by *P< 0.05 and **P< 0.01, based on the Wilcoxon rank-sum test comparing the SRCC of scTICG with each of the other methods.

Fig. 3 shows the performance of scTICG across the above four datasets. Overall, the results demonstrate that scTICG exhibits higher accuracy in pseudotime ordering compared to the competing methods. Specifically, For the HSMM dataset, the POS and SRCC scores for the pseudotime inferred by scTICG are 0.701 (95%CI: 0.668-0.735) and 0.677 (95%CI: 0.642-0.712), respectively, approximately 2.6% higher than those achieved by the sub-optimal method, DensityPath. On the Fibroblast dataset, scTICG achieves POS and SRCC scores of 0.972 (95%CI: 0.969-0.974) and 0.937 (95%CI: 0.934-0.940), respectively, slightly surpassing the scores of the closest competitor method, DensityPath. Regarding the HE dataset, scTICG achieves the highest POS (0.959, 95%CI: 0.955-0.963) and SRCC (0.918, 95%CI: 0.913-0.923) scores, which are about 2.7% and 3.2% higher than those of the sub-optimal method, TSCAN. On the iPSC dataset, scTICG also outperforms all competing methods, achieving the highest POS (0.974, 95%CI: 0.970-0.978) and SRCC (0.950, 95%CI: 0.943-0.957) scores. Overall, on the HE and iPSC datasets, scTICG significantly outperforms the compared methods. While on the HSMM and Fibroblast datasets, scTICG exhibits higher performance than the other methods, except DensityPath and scTite.

Next, to further validate the performance of scTICG on large-scale datasets with more complex trajectories, we apply it to the Hrpi dataset. According to the previous study [38], the Hrpi dataset predominantly features four distinct developmental trajectories. Specifically, the first trajectory encompasses the primed-reprogramming process, where fibroblast cells differentiate into primed iPSC cells. The second trajectory involves the differentiation from fibroblast to naive cells. The third trajectory captures the differentiation of the Trophectoderm branches during reprogramming. The fourth trajectory involves differentiation from fibroblasts to refractory cells. Inspired by the original paper, and considering that differentiation into refractory cells does not involve reprogramming, we focus on the first three differentiation trajectories of the Hrpi dataset, which comprises 27,901 cells. scTICG successfully infers a trajectory topology consistent with original studies (Fig. 4C), indicating its capability to accurately reconstruct complex cell development trajectories. In terms of accuracy, due to the absence of external information such as data collection time, the POS score for Hrpi could not be computed. Instead, we use the pseudotime inferred

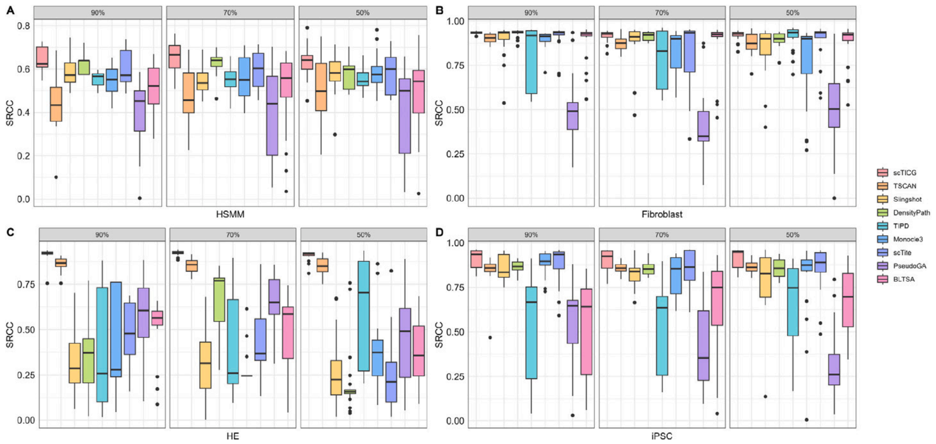


Fig. 5. Robustness analysis with 90%, 70% and 50% cell sampling rates with (A) HSMM, (B) Fibroblast, (C) HE and (D) iPSC.

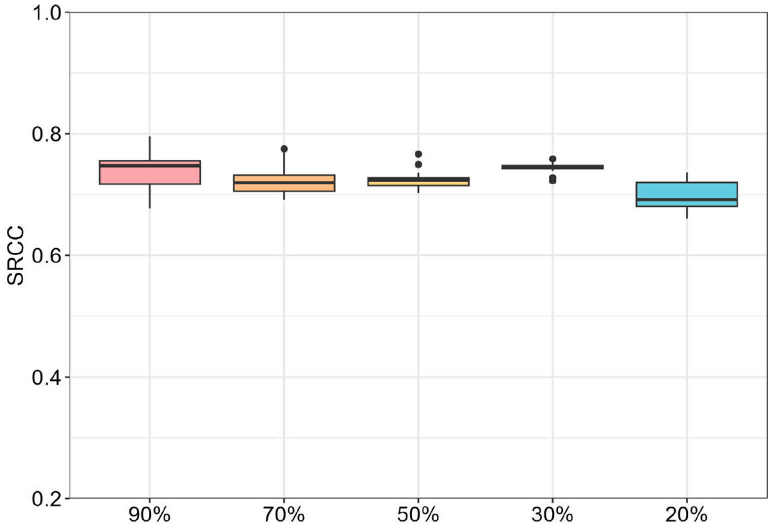


Fig. 6. Robustness analysis of scTICG with different cell sampling rates on Hrpi dataset.

by a combination of CytoTRACE and Monocle3 provided in [38] as the reference pseudotime, and calculate the SRCC score of the inferred pseudotime ordering against this reference. scTICG achieves statistically significantly higher SRCC score(0.689, $95\%CI$:0.681-0.698) than the competing methods, with an improvement of approximately 5.1% over the sub-optimal method scTite (Fig. 4D).

3.2. Robustness analysis

We compare the robustness of scTICG with other competing methods, evaluating SRCC scores under different disturbances on multiple datasets. Each method is assessed using the same subsamples, repeated 20 times. First, we test the robustness under different cell sampling rates, with 90%, 70% and 50% of cells randomly sampled from the original datasets (HSMM, Fibroblast, HE and iPSC). As shown in Fig. 5, in these four datasets, the SRCC scores of scTICG consistently exhibit minimal fluctuations, with average scores exceeding those of the eight other competing methods. Subsequently, we apply scTICG to the Hrpi dataset to further assess its robustness on more complex trajectories, the perturbed datasets are generated by randomly sampling 90%, 70%, 50%, 30% and 20% from Hrpi dataset. As shown in Fig. 6, the fluctuation range of the SRCC score of scTICG remains below 0.05 (90%: 0.740; 70%: 0.721; 50%: 0.724; 30%: 0.744; 20%: 0.697).

In addition, we analyze the robustness under different gene filtering ratios, varying from 10% to 20%. The experimental results presented in Fig. 7 demonstrate that the fluctuations in SRCC scores of scTICG across all three test datasets are small under different gene filtering ratios, which indicates that our method remains steady under different gene ratios. We further evaluate the robustness of scTICG with different dimensionality reduction methods. As shown in Supplementary Figure 1, scTICG maintains robust performance with nonlinear dimensionality reduction methods, while its performance slightly declines with linear methods such as PCA.

3.3. Scalability analysis

To evaluate the scalability of scTICG relative to other trajectory inference (TI) methods, we compare their running time across different datasets. The experiment is repeated 10 times for each dataset, and the average running time is recorded. The results presented in Fig. 8 illustrate the running times of these methods. Generally, scTICG demonstrates superior efficiency compared to the majority of competing methods. Moreover, scTICG enhances time efficiency by employing a subsampling approach for datasets with a large number of cells, such as Hrpi. Specifically, by using a heap to store the neighbor nodes of each cell, the time complexity of reconstructing cell development trajectory based on a greedy strategy is about $O(K \cdot n\log(n))$, where K is the number of clusters and n denotes the number of cells involved in each trajectory reconstruction. The computational burden increases as the number of cells grows. To address the challenge posed by the

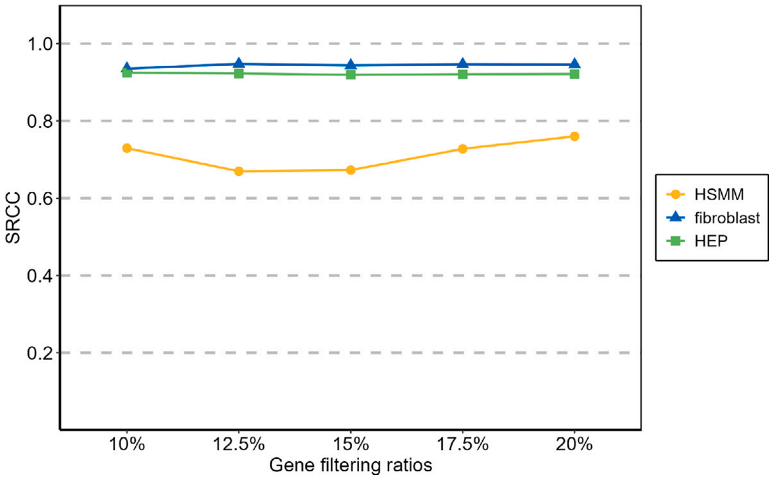
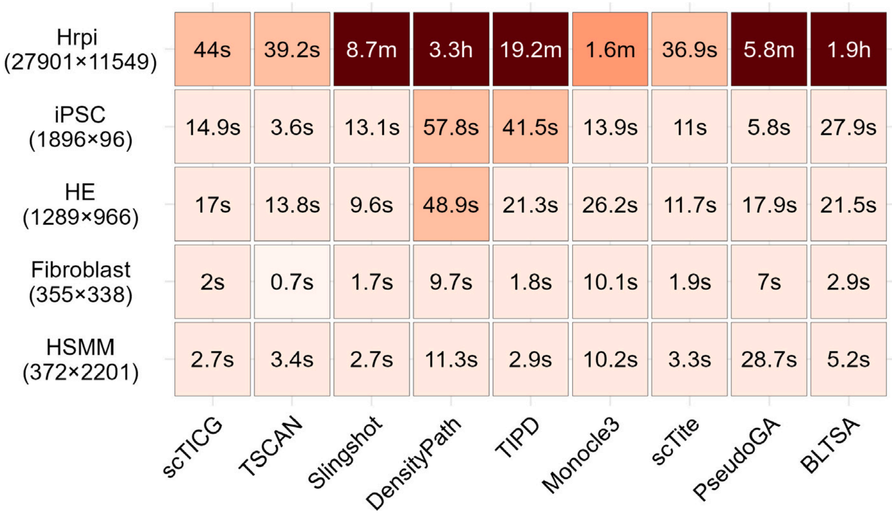


Fig. 7. Robustness analysis under different gene filtering ratios.



	scTICG	TSCAN	Slingshot	DensityPath	TIPD	Monocle3	scTite	PseudoGA	BLTSA
Hrpi (27901×11549)	44s	39.2s	8.7m	3.3h	19.2m	1.6m	36.9s	5.8m	1.9h
iPSC (1896×96)	14.9s	3.6s	13.1s	57.8s	41.5s	13.9s	11s	5.8s	27.9s
HE (1289×966)	17s	13.8s	9.6s	48.9s	21.3s	26.2s	11.7s	17.9s	21.5s
Fibroblast (355×338)	2s	0.7s	1.7s	9.7s	1.8s	10.1s	1.9s	7s	2.9s
HSMM (372×2201)	2.7s	3.4s	2.7s	11.3s	2.9s	10.2s	3.3s	28.7s	5.2s

Fig. 8. Comparison of running Time across different methods (*no. of cells*no. of features*).

increasing volume of single-cell datasets, we adopt sampling strategy for trajectory reconstruction.

3.4. Differential analysis

To further assess the performance of scTICG, we compare its accuracy in identifying marker genes with the competing trajectory inference methods on four real scRNA-seq datasets, including HSMM, Fibroblasts, HE and iPSC. First, differentially expressed genes are filtered using an absolute $\log_2$ (fold change) threshold of 1 and a p-value threshold of 0.05 to construct volcano plots, which showcase genes with significant expression changes during the cell development process. Next, for each dataset, we apply tradeSeq [45], a tool for analyzing gene expression along trajectories, to select five top differentially expressed genes along ground-truth cell development trajectory as gold standard genes from marker genes mentioned in original studies [36,37, 46,47], with details provided in Supplementary Table 4. Gold standard genes are genes that are known to be differentially expressed during the biological process [14], and they are marked in the volcano plots.

Overall, scTICG can effectively identify both up-regulated and down-regulated gold standard genes during cell development. In the dataset HSMM, *MYF5* is down-regulated during cell development, whereas *MYH3*, *ENO3*, *TNNT1* and *MEF2C* are up-regulated. All methods successfully capture the trends for these genes, except TSCAN, Slingshot, PseudoGA and BLTSA (Fig. 9). The volcano plot analyses for Fibroblast, HE and iPSC are respectively shown in Supplementary Figures 4, 5 and 6.

4. Discussion and conclusion

With the advancement of single-cell sequencing technologies, a variety of computational methods have been developed for trajectory inference. Traditional cluster-based methods often overlook the continuous nature of cellular developmental processes, treating them as a series of discrete states. In contrast, graph-based methods recognize this continuity but are susceptible to noise, which can distort the inferred trajectories. To overcome these limitations, we propose scTICG, a novel algorithm that integrates the strengths of both cluster-based and graph-based methods to more accurately infer the trajectory of cells. Initially, scTICG employs hierarchical clustering to partition cells into distinct states, forming a coarse-grained trajectory using an MST constructed from these clusters. This step establishes a backbone structure of the trajectory. Recognizing the continuous nature of cell dynamics, scTICG refines this trajectory by focusing on critical cells, including initial, terminal and intermediate cells. For each pair of directly connected cell clusters, a cell–cell graph is constructed. This graph is then pruned to remove noise, achieved by eliminating edges that do not align well with the overall trajectory, as determined by cosine similarity

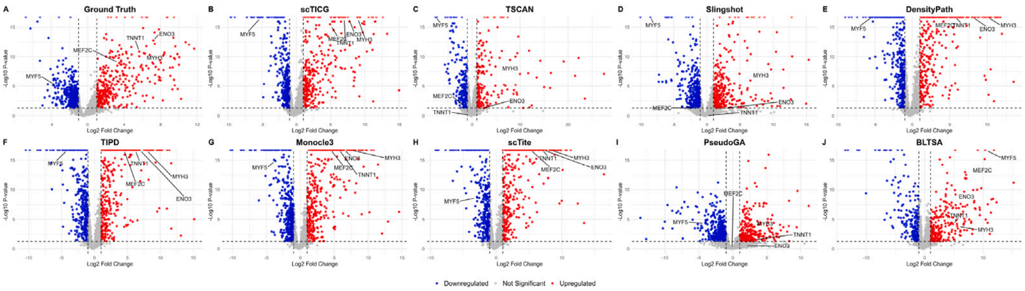


Fig. 9. Volcano plot of the HSMM dataset(Down-regulated genes: $\log_2$(fold change) < −1 and *p*-value < 0.05; up-regulated genes have a $\log_2$(fold change) > 1 and *p*-value < 0.05).

between cell connections and the cluster-to-cluster trajectory. Further refinement is achieved through the integration of the graph centrality algorithm, which helps identify critical cells within the cell–cell graph. These critical cells, particularly the initial and terminal ones, serve as anchors for reconstructing the trajectory using a greedy strategy, ensuring that the most probable developmental paths are followed. Finally, the principal curve algorithm is applied to smooth the refined trajectory. This smoothing step projects cells onto the trajectory, allowing for the calculation of pseudotime. However, the time complexity of trajectory reconstruction is approximately $O(K \cdot n\log(n))$. To address the challenge posed by large datasets, we adopt subsampling strategy for trajectory reconstruction. To verify the performance of scTICG in trajectory inference, we apply it to four real scRNA-seq datasets and one synthetic dataset, comparing its performance with that of eight state-of-the-art trajectory inference methods. The comparative results demonstrate that scTICG not only reconstructs cell trajectories with high accuracy but also consistently outperforms other methods under various evaluation metrics.

CRediT authorship contribution statement

Yanglan Gan: Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition. **Jiaqi Chu:** Writing – original draft, Methodology, Formal analysis, Data curation. **Guangwei Xu:** Validation, Supervision, Investigation, Formal analysis. **Cairong Yan:** Writing – review & editing, Validation. **Guobing Zou:** Writing – review & editing, Supervision, Investigation, Funding acquisition.

Funding

This work wa sponsored in part by the National Natural Science Foundation of China (62172088, 62272290) and Shanghai Natural Science Foundation, China (21ZR1400400).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neucom.2025.129482.

Data availability

The datasets utilized in this study are all publicly accessible. Here are the details regarding the sources of each dataset.

The HSMM dataset is available from the NCBI Gene Expression Omnibus (GEO) under the accession number GSE52529.

The Fibroblast dataset can be accessed under the accession number GSE67310.

The Hrpi dataset is available on the website http://hrpi.ddnetbio.com/.

The HE dataset can be downloaded from Zenodo at this link https://zenodo.org/records/1443566/files/real/gold/human-embryos_petropoulos.rds?download=1.

The iPSC dataset can be accessed from https://www.pnas.org/highwire/filestream/29285/field_highwire_adjunct_files/1/pnas.1621412114.sd02.xlsx.

References

[1] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W.M. Mauck, Y. Hao, M. Stoeckius, P. Smibert, R. Satija, Comprehensive integration of single-cell data, Cell 177 (2019) 1888–1902, http://dx.doi.org/10.1016/j.cell.2019.05.031.

[2] X. Ma, L. Yu, P. Wang, X. Yang, Discovering DNA methylation patterns for long non-coding RNAs associated with cancer subtypes, Comput. Biol. Chem. 69 (2017) 164–170, http://dx.doi.org/10.1016/j.compbiolchem.2017.03.014.

[3] X. Ma, W. Tang, P. Wang, X. Guo, L. Gao, Extracting stage-specific and dynamic modules through analyzing multiple networks associated with cancer progression, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (2016) 647–658, http://dx.doi.org/10.1109/TCBB.2016.2625791.

[4] X. Ma, P. Sun, G. Qin, Identifying condition-specific modules by clustering multiple networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (2017) 1636–1648, http://dx.doi.org/10.1109/TCBB.2017.2761339.

[5] A. Baysoy, Z. Bai, R. Satija, R. Fan, The technological landscape and applications of single-cell multi-omics, Nature Rev. Mol. Cell Biol. 24 (2023) 695–713, http://dx.doi.org/10.1038/s41580-023-00615-w.

[6] W. Wu, Z. Liu, X. Ma, jSRC: a flexible and accurate joint learning algorithm for clustering of single-cell RNA-sequencing data, Brief. Bioinform. 22 (2021) bbaa433, http://dx.doi.org/10.1093/bib/bbaa433.

[7] W. Saelens, R. Cannoodt, H. Todorov, Y. Saeys, A comparison of single-cell trajectory inference methods, Nature Biotechnol. 37 (2019) 547–554, http://dx.doi.org/10.1038/s41587-019-0071-9.

[8] R. Qu, X. Cheng, E. Sefik, J.S.S. III, B. Landa, F. Strino, S. Platt, J. Garritano, I.D. Odell, R. Coifman, et al., Gene trajectory inference for single-cell data by optimal transport metrics, Nature Biotechnol. (2024) 1–11, http://dx.doi.org/10.1038/s41587-024-02186-3.

[9] X. Dong, J.R. Leary, C. Yang, M.A. Brusko, T.M. Brusko, R. Bacher, Data-driven selection of analysis decisions in single-cell RNA-seq trajectory inference, Brief. Bioinform. 25 (2024) bbae216, http://dx.doi.org/10.1093/bib/bbae216.

[10] L. Deconinck, R. Cannoodt, W. Saelens, B. Deplancke, Y. Saeys, Recent advances in trajectory inference from single-cell omics data, Curr. Opin. Syst. Biol 27 (2021) 100344, http://dx.doi.org/10.1016/j.coisb.2021.05.005.

[11] X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H.A. Pliner, C. Trapnell, Reversed graph embedding resolves complex single-cell trajectories, Nature Methods 14 (2017) 979–982, http://dx.doi.org/10.1038/nmeth.4402.

[12] Z. Ji, H. Ji, TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis, Nucleic Acids Res. 44 (2016) e117, http://dx.doi.org/10.1093/nar/gkw430.

[13] K. Street, D. Risso, R.B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, S. Dudoit, Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics, BMC Genomics 19 (2018) 1–16, http://dx.doi.org/10.1186/s12864-018-4772-0.

[14] J. Xie, Y. Yin, J. Wang, TIPD: A probability distribution-based method for trajectory inference from single-cell RNA-seq data, Interdiscip. Sci. Comput. Life Sci. 13 (2021) 652–665, http://dx.doi.org/10.1007/s12539-021-00445-4.

[15] J. Cao, M. Spielmann, X. Qiu, X. Huang, D.M. Ibrahim, A.J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F.J. Steemers, et al., The single-cell transcriptional landscape of mammalian organogenesis, Nature 566 (2019) 496–502, http://dx.doi.org/10.1038/s41586-019-0969-x.

[16] L. Li, Y. Zhao, H. Li, S. Zhang, BLTSA: pseudotime prediction for single cells by branched local tangent space alignment, Bioinformatics 39 (2023) btad054, http://dx.doi.org/10.1093/bioinformatics/btad054.

[17] S.C. Bendall, K.L. Davis, E. ad David Amir, M.D. Tadmor, E.F. Simonds, T.J. Chen, D.K. Shenfeld, G.P. Nolan, D. Pe'er, Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development, Cell 157 (2014) 714–725, http://dx.doi.org/10.1016/j.cell.2014.04.005.

[18] L. Haghverdi, M. Büttner, F.A. Wolf, F. Buettner, F.J. Theis, Diffusion pseudotime robustly reconstructs lineage branching, Nature Methods 13 (2016) 845–848, http://dx.doi.org/10.1038/nmeth.3971.

[19] F.A. Wolf, F.K. Hamey, M. Plass, J. Solana, J.S. Dahlin, B. Göttgens, N. Rajewsky, L. Simon, F.J. Theis, PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells, Genome Biol. 20 (2019) 1–9, http://dx.doi.org/10.1186/s13059-019-1663-x.

[20] Y. Gan, C. Guo, W. Guo, G. Xu, G. Zou, Entropy-based inference of transition states and cellular trajectory for single-cell transcriptomics, Brief. Bioinform. 23 (2022) bbac225, http://dx.doi.org/10.1093/bib/bbac225.

[21] P. Zhou, S. Wang, T. Li, Q. Nie, Dissecting transition cells from single-cell transcriptome data through multiscale stochastic dynamics, Nat. Commun. 12 (2021) 5609, http://dx.doi.org/10.1038/s41467-021-25548-w.

[22] T. Lan, G. Hutvagner, X. Zhang, T. Liu, L. Wong, J. Li, Density-based detection of cell transition states to construct disparate and bifurcating trajectories, Nucleic Acids Res. 50 (2022) e122, http://dx.doi.org/10.1093/nar/gkac785.

[23] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Phil. Trans. R. Soc. A 374 (2016) 20150202, http://dx.doi.org/10.1098/rsta.2015.0202.

[24] L. der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008).

[25] D. Kobak, P. Berens, The art of using t-SNE for single-cell transcriptomics, Nat. Commun. 10 (2019) 5416, http://dx.doi.org/10.1038/s41467-019-13056-x.

[26] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2018, arXiv preprint arXiv:1802.03426.

[27] K.R. Moon, D.V. Dijk, Z. Wang, S. Gigante, D.B. Burkhardt, W.S. Chen, K. Yim, A. van den Elzen, M.J. Hirn, R.R. Coifman, et al., Visualizing structure and transitions in high-dimensional biological data, Nature Biotechnol. 37 (2019) 1482–1492, http://dx.doi.org/10.1038/s41587-019-0336-3.

[28] M. Charrad, N. Ghazzali, V. Boiteau, A. Niknafs, NbClust: an R package for determining the relevant number of clusters in a data set, J. Stat. Softw. 61 (2014) 1–36, http://dx.doi.org/10.18637/jss.v061.i06.

[29] W. Wu, X. Ma, Network-based structural learning nonnegative matrix factorization algorithm for clustering of scRNA-seq data, IEEE/ACM Trans. Comput. Biol. Bioinform. 20 (2022) 566–575, http://dx.doi.org/10.1109/TCBB.2022.3161131.

[30] M. Lange, V. Bergen, M. Klein, M. Setty, B. Reuter, M. Bakhti, H. Lickert, M. Ansari, J. Schniering, H.B. Schiller, et al., CellRank for directed single-cell fate mapping, Nature Methods 19 (2022) 159–170, http://dx.doi.org/10.1038/s41592-021-01346-6.

[31] J. Zhang, Y. Luo, Degree centrality, betweenness centrality, and closeness centrality in social network, in: 2017 2nd International Conference on Modelling, Simulation and Applied Mathematics, MSAM2017, 2017, pp. 300–303, http://dx.doi.org/10.2991/msam-17.2017.68.

[32] S. Tabassum, F.S.F. Pereira, S. Fernandes, J. Gama, Social network analysis: An overview, Wiley Interdiscip. Reviews Data Min. Knowl. Discov. 8 (2018) e1256, http://dx.doi.org/10.1002/widm.1256.

[33] E.K. Mbaru, M.L. Barnes, Key players in conservation diffusion: Using social network analysis to identify critical injection points, Biol. Cons. 210 (2017) 222–232, http://dx.doi.org/10.1016/j.biocon.2017.03.031.

[34] L. Scrucca, M. Fop, T.B. Murphy, A.E. Raftery, mclust 5: clustering, classification and density estimation using Gaussian finite mixture models, R J. 8 (2016) 289, http://dx.doi.org/10.32614/rj-2016-021.

[35] T. Hastie, W. Stuetzle, Principal curves, J. Amer. Statist. Assoc. 84 (1989) 502–516, http://dx.doi.org/10.1080/01621459.1989.10478797.

[36] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N.J. Lennon, K.J. Livak, T.S. Mikkelsen, J.L. Rinn, The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells, Nature Biotechnol. 32 (2014) 381–386, http://dx.doi.org/10.1038/nbt.2859.

[37] B. Treutlein, Q.Y. Lee, J.G. Camp, M. Mall, W. Koh, S.A.M. Shariati, S. Sim, N.F. Neff, J.M. Skotheim, M. Wernig, et al., Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq, Nature 534 (2016) 391–395, http://dx.doi.org/10.1038/nature18323.

[38] X. Liu, J.F. Ouyang, F.J. Rossello, J.P. Tan, K.C. Davidson, D.S. Valdes, J. Schröder, Y.B.Y. Sun, J. Chen, A.S. Knaupp, et al., Reprogramming roadmap reveals route to human induced trophoblast stem cells, Nature 586 (2020) 101–107, http://dx.doi.org/10.1038/s41586-020-2734-6.

[39] D. van Dijk, J. Nainys, R. Sharma, P. Kaithail, A.J. Carr, K.R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy, D. Pe'er, MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data, BioRxiv (2017) 111591, http://dx.doi.org/10.1101/111591.

[40] W. Gong, I.-Y. Kwak, P. Pota, N. Koyano-Nakagawa, D.J. Garry, Drimpute: imputing dropout events in single cell RNA sequencing data, BMC Bioinformatics 19 (2018) 1–10, http://dx.doi.org/10.1186/s12859-018-2226-y.

[41] G.C. Linderman, J. Zhao, M. Roulis, P. Bielecki, R.A. Flavell, B. Nadler, Y. Kluger, Zero-preserving imputation of single-cell RNA-seq data, Nat. Commun. 13 (2022) 192, http://dx.doi.org/10.1038/s41467-021-27729-z.

[42] D. Tran, B. Tran, H. Nguyen, T. Nguyen, A novel method for single-cell data imputation using subspace regression, Sci. Rep. 12 (2022) 2697, http://dx.doi.org/10.1038/s41598-022-06500-4.

[43] Z. Chen, S. An, X. Bai, F. Gong, L. Ma, L. Wan, DensityPath: an algorithm to visualize and reconstruct cell state-transition path on density landscape for single-cell RNA sequencing data, Bioinformatics 35 (2019) 2593–2601, http://dx.doi.org/10.1093/bioinformatics/bty1009.

[44] P.K. Mondal, U.S. Saha, I. Mukhopadhyay, Pseudoga: cell pseudotime reconstruction based on genetic algorithm, Nucleic Acids Res. 49 (2021) 7909–7924, http://dx.doi.org/10.1093/nar/gkab457.

[45] K. den Berge, H. de Bézieux, K. Street, W. Saelens, R. Cannoodt, Y. Saeys, S. Dudoit, L. Clement, Trajectory-based differential expression analysis for single-cell sequencing data, Nat. Commun. 11 (2020) 1201, http://dx.doi.org/10.1038/s41467-020-14766-3.

[46] S. Petropoulos, D. Edsgärd, B. Reinius, Q. Deng, S.P. Panula, S. Codeluppi, A.P. Reyes, S. Linnarsson, R. Sandberg, F. Lanner, Single-cell RNA-seq reveals lineage and x chromosome dynamics in human preimplantation embryos, Cell 165 (2016) 1012–1026, http://dx.doi.org/10.1016/j.cell.2016.03.023.

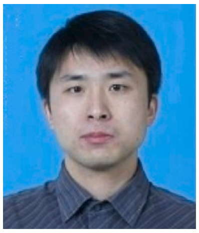
[47] R. Bargaje, K. Trachana, M.N. Shelton, C.S. McGinnis, J.X. Zhou, C. Chadick, S. Cook, C. Cavanaugh, S. Huang, L. Hood, Cell population structure prior to bifurcation predicts efficiency of directed differentiation in human induced pluripotent cells, Proc. Natl. Acad. Sci. 114 (2017) 2271–2276, http://dx.doi.org/10.1073/pnas.1621412114.

Yanglan Gan is a professor in the school of computer science and technology at Donghua University, Shanghai, China. She received the Ph.D. degree in computer science from Tongji University in 2012, China. She has worked as a Visiting Scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. Her research interests include bioinformatics, data mining, and Web services. She has published more than 60 papers on international journals and conferences, including Briefings in Bioinformatics, IEEE/ACM TCBB, Bioinformatics and Knowledge-based Systems. She served as a program committee member on BIBM from 2018 to 2024.

Jiaji Chu is currently a master's student at the School of Computer Science and Technology, Donghua University, China. Before that, she received a Bachelor's degree from Zhejiang Normal University in 2022. Her research interests include data mining and bioinformatics.

Guangwei Xu is Professor in the School of Computer Science and Technology, Donghua University, China. He received his Ph.D. in Computer Science from Tongji University, Shanghai, China, 2003. His current research interests focus on intelligent big data analysis and computing security. He has published around 60 papers on premier international journals and conferences, including Computer Networks, Journal of Parallel and Distributed Computing and Briefings in Bioinformatics.

Cairog Yan is Professor in the School of Computer Science and Technology, Donghua University, China. She received his Ph.D. in Computer Science from Xi'an Jiaotong University, Xi'an, China, 2006. Her current research interests focus on recommender systems, big data analysis and distributed parallel computing. She has published around 60 papers on premier international journals and conferences, including Applied Intelligence, Knowledge-Based Systems, International Journal of Intelligent Systems and Bioinformatics.



Guobing Zou is Professor in the School of Computer Engineering and Science, Shanghai University, China. He received his Ph.D. in Computer Science from Tongji University, Shanghai, China, 2012. His current research interests focus on data mining, intelligent algorithms and services computing. He has published around 90 papers on premier international journals and conferences, including Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, IEEE Transactions on Services Computing, AAAI, ICWS and ICSOC.